



Conociendo JSTL



Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas una vez más a este curso de Servlets. En el video pasado vimos cómo aplicar expression languages para poder disminuir el código que utilizamos para reemplazar nuestros scriptlets con códigos más pequeños y les dije que íbamos a reemplazar este lazo for o este bucle for que nosotros tenemos acá dentro de nuestras listas de empresas,

[00:26] Pero la cuestión es que para poder hacer eso, nosotros lo que vamos a necesitar antes es importar una biblioteca. ¿Por qué? Porque nuestro Tomcat por default no sabe cómo reemplazar ese foro utilizando lo que nosotros vamos a llamar JSTL. Entonces para eso yo ya bajé acá las bibliotecas, que ustedes las van a tener ahí en los links del curso.

[00:57] Ustedes lo que tienen que hacer es copiar esos archivos, o pueden arrastrarlo si quieren hasta esta carpeta. Entonces, dentro de webapp, WEB-INF, tenemos una carpeta ya creada llamada lib, entonces ahí dentro ustedes sueltan los archivos, copian. Vean que nuestro Eclipse va a intentar recargar nuestro proyecto automáticamente. Vean que incluso se reinició también el servidor.

[01:30] Entonces solo por las dudas clic derecho en nuestro proyecto gerenciador, refresh y bien, con esto nosotros tendremos nuestra biblioteca JSTL. Ahora, ¿cómo hacemos para reemplazar este for? Bien, lo que nosotros tenemos que hacer es lo siguiente. Voy a bajar un poco acá para poder tener

espacio para poder ver el código anterior, vamos a abrir y cerrar nuestra etiqueta `ul`.

[01:59] Y acá lo que nosotros vamos a utilizar es una tag, así como nosotros en HTML dentro de nuestro JSP utilizamos tags o etiquetas, nosotros también vamos a usar etiquetas para utilizar este `for` de nuestra biblioteca JSTL. Entonces para eso vamos a abrir una etiqueta y vamos a colocar la palabra `forEach`, vamos a cerrar y vamos a cerrar nuestro `forEach`.

[02:25] ¿Por qué `forEach` y no `for`? Bueno, la cuestión es la siguiente. Anteriormente o incluso dentro de Java, nosotros utilizamos el `for`, por ejemplo, colocando `for` declaramos una variable `Int i = 0, i < tal cosa, i++`. Nuestro `forEach` sirve para que nosotros podamos enviarle directamente una lista y a cada uno de los elementos de esa lista le coloca un nombre y en cada iteración se utiliza ese nombre para poder acceder a algún atributo de ese objeto.

[03:00] Por ejemplo acá en este `for` dentro de esta lista nosotros utilizamos un `forEach` acá, en realidad. Acá nosotros tenemos ese `for`, tenemos una lista y a cada elemento de esa lista le asigna el nombre `empresa` y en cada iteración va a ir moviéndose entre cada uno de los elementos. Entonces utilizamos esa `empresa` para poder obtener el nombre de una empresa en particular dentro de esa lista, dentro de cada iteración.

[03:28] Entonces acá utilizamos ese `forEach`. La cuestión es la siguiente, nosotros tenemos que decirle cómo hacer de dónde viene ese `forEach`, porque no sabe de dónde tiene que obtener esa tag. Nos va a decir acá que no conoce esa tag `forEach`, entonces, lo que tenemos que hacer es lo siguiente.

[03:49] Así como nosotros utilizábamos esta sintaxis para poder importar nuestra biblioteca, usábamos esa sintaxis de scriptlet más un `@` y acá colocamos la palabra `page`, que es decir, lo que vamos a colocar acá va a afectar

a nuestra página. Nosotros lo que tenemos que utilizar acá es el nombre taglib. ¿Por qué taglib?

[04:13] Porque nosotros vamos a utilizar una librería que sirve para colocar tag dentro de nuestro HTML, JSP en este caso, entonces nosotros tenemos que usar esa sintaxis scriptlet @ taglib. Luego de eso, tenemos que decirle cuál es el nombre de esa librería que vamos a utilizar para las tags.

[04:36] Entonces colocan “Ctrl + espacio” acá y utilizan uri. ¿Qué es esa uri? Ese es el nombre de nuestra biblioteca, pero no se confundan, no tiene que colocar el nombre de acá de estos archivos, sino que es el nombre que existe dentro de esos .jar existe un nombre específico en el cual está nuestro JSP, va a buscar y va a encontrar esa librería ahí dentro de ese jar.

[05:09] Entonces esa librería es la siguiente: “http: //java.sun.com/”. Si aprietan “Ctrl + espacio”, ya les va a dar algunas sugerencias, si a ustedes no les aparece no se preocupen porque puede ser que Eclipse no haya recargado bien y no haya identificado aun bien las librerías. Sin embargo, si ustedes escriben lo que yo voy a colocar acá les va a funcionar perfecto.

[05:44] Entonces colocamos luego de la barra, jsp, porque estamos dentro de un archivo JSP, para que vayan acordándose de cómo es el camino, /jstl, porque viene de Java Standard Taglib. Ese es el nombre del JSTL, Java Standard Taglib. Y dentro de ese JSTL, vamos a usar la biblioteca core, el archivo core.

[06:19] Bien, eso es lo necesario, mínimo y necesario para nosotros poder utilizar nuestro forEach. Pero además de eso, además de colocar una uri ustedes tienen que colocar un prefijo. Ya les voy a explicar para qué sirve. Vamos a hacer “Ctrl + espacio” y colocan prefix. Eso ya automáticamente les coloca eso y acá pueden poner el nombre que ustedes quieren.

[06:44] Pueden poner “Juanito” si quisieran, Juanito, pero no. O sea, la idea es colocar un nombre acorde. ¿Acorde a qué? Este prefijo nos va a ayudar, para

que nosotros podamos decir a la tag: “busca en esta librería, Juanito, en este caso, y dentro de esa librería usá el forEach”.

[07:10] La cuestión es que no vamos a dejar acá el nombre Juanito. Vamos a colocar, por ejemplo “c” que es el estándar, normalmente dentro del mundo de Servlets y de JSTL se utiliza “c” porque “c” viene de core. Entonces vamos a utilizar el prefijo “c”. ¿Y cómo utilizamos esto? Nosotros tenemos que colocar acá c: y ahí nos queda el forEach.

[07:33] Recuerden, ese “c” viene del prefijo que es el que nos va a apuntar a nuestra biblioteca, entonces vamos a usar forEach de nuestra biblioteca. Perfecto ya nos está dando un error, incluso nos está diciendo que ya reconoce ese forEach y nos está diciendo que faltan argumentos. ¿Qué argumentos vamos a utilizar?

[07:54] Vean lo siguiente. En nuestro for nosotros utilizamos dos cosas principalmente. Utilizamos una lista que viene de nuestro dispatcher, de nuestro Servlet que enviamos un dispatcher y esa lista está con el nombre de empresas, entonces nosotros utilizamos esa lista que se llama “empresas” y luego utilizamos una empresa que está dentro de esa lista.

[08:20] Entonces, ¿cómo haríamos para obtener esa lista utilizando expression languages? Nosotros utilizaríamos, por ejemplo, este signo de dólar, abrimos y cerramos llaves, empresas porque ese es el nombre con el que viene dentro de nuestra request. Entonces vamos a usar eso.

[08:40] Vamos a usar, vamos a hacer un “Ctrl + X” y acá vamos a hacer un “Ctrl - espacio” y colocamos items, items es la lista de items por la que tiene que iterar. Entonces acá vamos a hacer un "Ctrl + V" en nuestro expression languages que habíamos copiado y vean que funciona muy bien, o sea, funciona muy bien esta mixtura entre expression language y nuestra biblioteca JSTL. Perfecto.

[09:09] Ya tenemos la lista. Ahora lo que nos falta es dar un nombre para cada uno de esos items. Para eso vamos a hacer un “Ctrl + espacio” de nuevo y vamos a hacer doble clic en ese var, ese var va a ser el nombre que nosotros vamos a usar, que es en este caso “empresa” que es el nombre que habíamos puesto acá dentro de ese for.

[09:30] Perfecto, ya tenemos un for. Ahora acá en la tag de cierre también tenemos que poner el “c:” que ahí se nos fue el error y ya tenemos nuestra parte del for. Tenemos esto acá y tenemos el cierre. Ahora lo que nos falta es colocar cada uno de los ítems que van a ir dentro de ese for, entonces para eso vamos a copiar lo que ya teníamos y pegar.

[09:58] Ahora, en vez de usar Scriptlets, vamos a utilizar expression languages. Entonces acá lo que vamos a hacer es cerrar, eliminar todo esto y vamos a usar expression languages, perdón, que también ahí. ¿Y acá vamos a colocar qué cosa? Empresas, que es el nombre de cada uno de los objetos que está dentro de esa lista de empresas con S.

[10:28] Entonces vamos a utilizar empresa.getNombre. Esto es lo que haríamos normalmente, pero expression languages nos facilita un poco todo esto y nos va a decir directamente que no necesitamos colocar los paréntesis, no necesitamos colocar ese get y ese nombre lo vamos a colocar en minúscula.

[10:49] Ya automáticamente comprende qué es lo que nosotros queremos hacer y por eso no es necesario hacer ese get al nombre con paréntesis.

Directamente colocando .nombre ya entiende que tiene que hacer un get a ese atributo. Vamos a dar un “Ctrl + S” y ya con esto nosotros ya podemos eliminar nuestro ul. Vean que tenemos un error acá, pero si eliminamos nuestro ul antiguo ya funciona.

[11:14] “Ctrl + S”, perfecto, sé que fueron muchas cosas, sé que es un poco complicado de comprender al comienzo. Pueden ver de nuevo este video para

comprenderlo realmente bien, pero vean que quedó mucho más simplificado de lo que teníamos anteriormente.

[11:33] O sea, comparado con lo que teníamos antes quedó mucho más simplificado y mucho más acorde usando etiquetas, como es normalmente en HTML y JSP. Perfecto. Con esto vamos a reiniciar el servidor, ya que importamos bibliotecas, hicimos varias cosas en la configuración, entonces mejor reiniciamos. Vamos a la consola, bien, no hay ningún error por ahora. Vamos a ir a nuestro Servlet listaEmpresas, el nombre de nuestro Servlet es lista empresas y vamos a dar un enter.

[12:08] Vamos a ver. Parece que todo funcionó normalmente vamos a inspeccionar el código y vean que ya no tenemos nada de los scriplets de import. No tenemos nada de nuestra Taglib, entonces quedó bonito. Con esto ya tenemos un JSP, un JSTL bastante profesional.

[12:35] En el día a día cuando ustedes trabajen van a trabajar con estas